DOCUMENTAȚIA PROIECTULUI

**Magazin Componente PC**

STUDENȚI

Miron Ioana Codruta

Margarit Darius Armand

**Cuprins**

1. **Prezentarea proiectului**
2. **Tehnologiile folosite**
3. **Baza de date**
4. **Prezentarea API-ului**
5. **Prezentare despre cum poate fi utilizata aplicatia**
6. **Concluzii si contributii**
7. **Link GIT catre codul proiectului**

***1. Prezentarea proiectului***

Proiectul nostru se concentrează pe dezvoltarea unei platforme administrative pentru gestionarea unui magazin online de componente PC. Aceasta permite utilizatorilor să se autentifice și să acceseze un mediu securizat, unde pot adăuga, actualiza și șterge produse și producători. În plus, administratorii au capacitatea de a gestiona roluri și utilizatori, inclusiv adăugarea, actualizarea și ștergerea acestora. Platforma este esențială pentru a menține catalogul de produse actualizat și pentru a facilita managementul informațiilor despre produse și producători. Prin această soluție, se abordează problemele legate de administrarea eficientă a stocurilor și de actualizarea rapidă a informațiilor, asigurând o gestionare optimă a magazinului online.

***2. Tehnologiile folosite***

Proiectul nostru utilizează tehnologiile C# și .NET(Entity Framework) pentru dezvoltarea aplicației și SQL Server Management Studio (SSMS) pentru gestionarea bazei de date. În plus, implementăm migrațiile pentru a facilita versionarea și actualizarea structurii bazei de date.

***3. Baza de date***

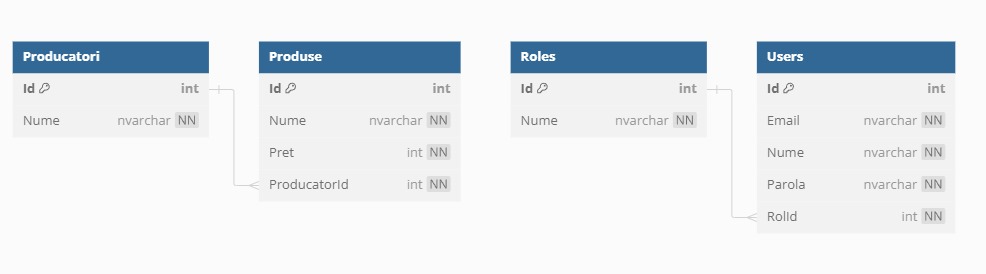


Diagrama prezintă patru tabele și relațiile dintre ele: Producatori cu coloanele Id (cheie primară) și Nume; Produse cu coloanele Id (cheie primară), Nume, Pret și ProducatorId (cheie externă către Producatori.Id), indicând o relație many-to-one; Roles cu coloanele Id (cheie primară) și Nume; și Users cu coloanele Id (cheie primară), Email, Nume, Parola și RolId (cheie externă către Roles.Id), indicând o altă relație many-to-one.

***4. Prezentarea API-ului***

A screenshot of a computer

Description automatically generated

A colorful lines on a white background

Description automatically generated

ProducatoriController gestionează operațiile CRUD pentru producători: citirea listei de producători (GET /api/producatori), adăugarea unui nou producător (POST /api/producatori/add), actualizarea informațiilor unui producător existent (PUT /api/producatori/{producatorId}/update) și ștergerea unui producător (DELETE /api/producatori/delete-producator).

ProdusController realizează operații similare pentru produse: citirea produselor (GET /api/produse), adăugarea unui produs (POST /api/produse/add), actualizarea unui produs (PUT /api/produse/{produsId}/edit) și ștergerea unui produs (DELETE /api/produse/delete-produs).

RolController administrează rolurile, permițând citirea (GET /api/rol), crearea (POST /api/rol), actualizarea (PUT /api/rol/{id}) și ștergerea rolurilor (DELETE /api/rol/{id}).

UserController se ocupă de utilizatori, incluzând înregistrarea (POST /register), autentificarea (POST /login), citirea tuturor utilizatorilor (GET /api/user), actualizarea (PUT /api/user/{id}) și ștergerea utilizatorilor (DELETE /api/user/{id}).

***5. Prezentare despre cum poate fi utilizata aplicatia***

Aplicatia contine 4 Controllere (User, Rol, Producatori, Produse) fiecare cu endpoint-urile sale pentru toate operatiile CRUD. Exista doua roluri care pot si asignate (User si Admin). Un admin are acces la toate endpoint-urile, pe cand un user are acces doar la cele din controllere-le Produse si Producatori. Un utilizator anonim (care nu s-a logat) poate accesa endpoint-urile de register si login pentru a-si face un cont sau a se conecta la unul existent.

În urma logării se genereaza un token care trebuie introdus în textbox-ul Authorize

(bearer token).Apoi, in functie de rolul fiecaruia utilizatorii pot accesa endpoint-urile.

***6. Concluzii si contributii***

**Contributii:**

* Codruta: autentificare/logare, autorizare(si tratarea exceptiilor pentru acestea)
* Darius: Controller-ele Produs,Producator(si tratarea exceptiilor pentru aceste)

**Concluzii:**

* Am invatat sa folosim limbajul C# si Entity Framework.
* Am invatat sa generam o baza de date folosind migratii
* Am invatat sa cream si sa testam un API
* Am invatat sa structuram un proiect in .NET

***7. Link GIT catre codul proiectului***

* <https://github.com/DariusMargarit/Tema_dotNet>